

# Project One, CS444, Spring 2018, Homework Group 10

Kamron Ebrahimi & Maximillian Schmidt & Brendan Byers  
ebrahimk, schmidtM, byersb

April 11, 2018

## **Abstract**

This report details the process of building the Linux Kernel and running the kernel on the Oregon State University, operating systems II server. To accomplish this task, the command line based, virtual machine, QEMU, was cloned and configured in order to produce a safe, experimental environment in which the Linux Kernel can be modified. This paper will walk the reader through the key components of configuring this environment.

# Contents

<b>1</b>	<b>Commands Used</b>	<b>2</b>
<b>2</b>	<b>QEMU Command Explanation</b>	<b>3</b>

# 1 Commands Used

In order to run the Linux Kernel in a QEMU virtual machine a folder titled "10" was created with the following file path /scratch/spring2018/10. A copy of the linux-yocto repository was then cloned and unzipped to the group 10 folder using the following commands from the group 10 folder:

- **wget http://git.yoctoproject.org/cgi.cgi/linux-yocto/snapshot/linux-yocto-3.19.2.zip**
- **unzip -q linux-yocto-3.19.2.zip**

With the linux-yocto repository successfully cloned the QEMU environment needed to be set up and run. Some preliminary measures were required to ensure this was performed properly. First the a script located at the file path /scratch/opt/environment-setup-i586-poky-linux was sourced. All members of group ten operate in the bash shell, thus this script at the aforementioned file path was used. To source this script the following command was executed from the root directory of os2.engr.oregonstate.edu.

- **source /scratch/opt/environment-setup-i586-poky-linux**

In order to ensure that the Makefile associated with the Linux Kernel executed properly the file located at the file path /scratch/files/config-3.19.2-yocto-standard was copied to a newly created folder named ".config" located within the linux-yocto-3.19.2 folder. Additionally the starting Kernel image, bzImage-qemux86.bin and drive file core-image-lsb-sdk-qemux86.ext4 were copied within the linux-yocto-3.19.2 folder. To accomplish this the following commands were executed from within the linux-yocto-3.19.2 folder.

- **cp /scratch/files/config-3.19.2-yocto-standard ./config**
- **cp /scratch/files/bzImage-qemux86.bin .**
- **cp /scratch/files/core-image-lsb-sdk-qemux86.ext4 .**

With the aforementioned preliminary configurations complete, the clone of the Linux Kernel could be compiled. The following command allocates four threads to build the Linux Kernel. A limit of the number of threads is used to ensure that no one group consumes too much computational resources on the shared os2.engr.oregonstate.edu server.

- **make -j4 all**

With the Kernel built, the last step was to boot the Kernel from within an instance of the QEMU virtual machine. First an instance of QEMU was executed on port 5510 of the os2 server using the following command:

- **qemu-system-i386 -gdb tcp::5510 -S -nographic -kernel bzImage-qemux86.bin -drive file=core-image-lsb-sdk-qemux86.ext4,if=virtio-enable-kvm -net none -usb -localtime -no-reboot -append "root=/dev/vda rw console=ttyS0 debug"**

Using GDB from another terminal window the Kernel could be executed. To complete this a group member had to connect via port 5510 via GDB to the QEMU virtual machine and execute the "vmlinux" process from within the linux-yocto-3.19.2 folder. The following chain of commands were used while an instance QEMU was running on port 5510, these commands were executed from within the group 10 folder.

- **\$GDB linux-yocto-3.14/vmlinux**

- **target remote :5510**
- **continue**

The above commands create a gdb connection to port 5510 of the os2 server and enters the vmlinux file. The vmlinux executable when run hoists an instance of the Linux Kernel from within the QEMU virtual machine. By running the "continue" command from within the gdb connection, vmlinux is executed which boots the Kernel from within QEMU.

## 2 QEMU Command Explanation